基于网络拓扑的公交查询方案

摘　要

公交、地铁线路和站点组成了一个极其复杂的网络结构，如何从这个网络的任意两个节点找到一条最优的乘车方案，传统遍历算法是很费时甚至不可行的，必须采取一种高效的方法。本文运用了网络拓扑的知识来分析问题，结合隐含枚举，双向搜索遍历，动态规划方法减少运算量，较好的解决了这一问题。

对于问题一，我们采用了网络拓扑进行分析，采用隐含枚举，双向搜索的方法，建立了两点之间线路搜索的动态规划多目标模型，设计了基于直达站点间点—点最优距离的广度优先搜索算法，得出了较好的结果，如：

$$S3359 \to S1828\text{：}S3359 \xrightarrow[31]{L436} S1784 \xrightarrow[1]{L176} S1828$$

$$S3359 \xrightarrow[4]{L15} S1327 \xrightarrow[13]{L201} S1790 \xrightarrow[5]{L41} S1828$$

对于问题二，我们在问题一已经给出的纯公交路径基础上，采取了增加地铁连通站点集合（两两可达）的方法，建立了求经地铁中转的最优线路的多目标模型，设计了基于搜索地铁出入站点的最优路径算法，得到了令人满意的结果，如：

$$S0087 \to S3676\text{：}S0087 \to D27 \xrightarrow[8]{T2} D36 \to S3676$$

对于问题三，我们采用了网络拓扑进行分析，确立了两点之间的距离正比于步行时间的原则，在此基础上，建立了基于归并相邻站点的最优线路的改良模型。

综合我们使用的各种方法，可以把原来很难实现的求解过程复杂度缩小数个数量级，使算法可行并可以搜索更多的区域，最终得到了令人满意的路径。

关键词：网络拓扑 隐含遍历 动态规划 点—点最优距离 广度优先搜索 最优路径

1. 问题提出与分析

2008年奥运会在京举行期间，将有大量游客到北京，北京公共交通系统的发展极大的满足了游客们在京的出行需求，同时也产生了多条公交线路的选择问题。为方便游客在京出行，需要一个便利的公交线路查询系统。当游客输入起始公交站点时，系统给出省时、省钱或者换乘少的线路。

为解决此问题，我们就应该了解题目中涉及的部分公交网络的特点：

(1)方向性：对于地铁，可知其所有站点是两两可达的，因此我们可以暂不考虑其方向性，而公交线路的运行是有方向的，一条公交线路有上行和下行两个行驶方向，即从始发站到终点站和从终点站回到始发站。部分公交线路上下行站点不变，但上下行站点分别位于道路两侧且位置并不重叠，另外一些公交线路上下行路线不完全相同，因此处理时应该分别对待。

(2)换车存在多可行站点：不同的公交线路在行程上有重叠以致多次重叠，在实际通行可能要求在换车以到达目的地，换车往往可以在两条路线的多个站点上进行，这可以进行一定的合并以减少运算的复杂度而不会对结果造成不良的影响。

(3)最优路径定义：最优路径应该定义为最短时间，最少费用，最小转车次数或其组成的某种函数（如带权值的函数）。交通网络上的最短路径和本题中线路的最短路径的意义也不相同。道路网络的最短路径即两点之间路径最短。而问题中除了通行耗时外，换车活动是有消耗的，不能完全为寻找最短的路径距离而随意换车。试想乘客每到一个公交站点都考虑换乘，才可以计算最短路径。但结果可能是：从A站到B站需要转好几次车或十几次车才能到达，这样的计算结果是没有什么实际意义的。

而已知的信息表明：只考虑公汽线路时，不需要考虑节点的抽象和连通性的问题，只要在区分上下行的要求下找到最优的线路即可。

如果同时考虑地铁线路和步行转站，则可在公汽线路选择的基础上将通过地铁或者步行换乘公汽的过程视为对公交线路站点的抽象。当然，这一过程还会受到各种其他因素，如地铁的车票价格，又如步行时长等因素的制约。

2. 符号系统

符号	符号意义	备注
T_0	起始站点	
T_e	终点	
n	某路经换乘次数	
T_i	从起点起，路经经过的第 i 个站点	i=1,2,3…n
E_i	从起点起，第 i 个节点可以选择的全部线路集合	i=0,1,2,3…n
e_i	路径上从起点起，第 i 个中转站点到下一个中转站点间的路线(i=n 时为到终点的路线)	i=0,1,2,3…n
(T_x, T_y)	任意两个站点 T_x 、T_y 之间的线路集合	
$Dis(T_x, T_y)$	任意两个站点 T_x 、T_y 之间的步行距离	单位：千米

λ_f	某游客对步行的不满意度	
$Count_i$	路径上从起点起，e_i 经过的站点数	i=0,1,2,3…n
$Time_i$	路径上从起点起，经过 e_i 所用时间	i=0,1,2,3…n 单位：分钟
$Cost_i$	路径上从起点起，经过 e_i 所用的金钱	i=0,1,2,3…n 单位：元
w	无量纲条件下的综合目标函数值	
λ_x	换乘的耗费，x=1,2,3,4 分别表示汽—汽、铁—铁、铁—汽、汽—铁转换耗费时间	单位：分钟

3. 模型建立

3.1 模型假设

1.所有相邻站点之间的公交线路耗时均为3分钟；
2.游客希望在尽可能少的换乘情况下抵达目标站点；
3.不考虑不同时间段的公路交通负荷的不同；
4.不考虑车辆在不同站点的拥挤程度不同；

3.2 模型建立

3.2.1 问题一，仅有公交车的线路搜索

我们的模型主要考虑的是给游客提供一条便捷可靠的公交线路，包括换乘的线路，在问题一中，将只考虑公交线路的处理，并建立一个多目标规划模型：

花费时间最少：

$$\min \quad \sum_{i=0}^{n} Time_i + (n-1)\times\lambda_1 \tag{1}$$

花费金钱最少：

$$\min \quad \sum_{i=0}^{n} Cost_i \tag{2}$$

中转次数最少：

$$\min \quad n \tag{3}$$

由于题目中要求因查询者的不同需求给出路线则我们综合起来，可以因乘客对时间，金钱，中转次数的不同需求把上述目标函数分别建立模型，即给出较优的几个解，然后结合其他目标进行进一步的优化，比如我们求得几条路径均花费最少时间，就可以用其他两个目标进行进一步的筛选。

约束条件是每次选择的线路，必须是经过该路段而且可以到达路段终点即下一中转站或终点的路径，即：

$$\begin{aligned} & e_i \in E_i \\ & e_i \in (T_i, T_{i+1}) \qquad i = 0,1,2...n \\ & T_{n+1} = T_e \end{aligned} \tag{4}$$

可以得出如下三个模型：

时间最优模型：

$$\min \quad \sum_{i=0}^{n} Time_i + (n-1)\times\lambda_1$$
$$s.t \quad \begin{cases} e_i \in E_i \\ e_i \in (T_i, T_{i+1}) \qquad i = 0,1,2...n \\ T_{n+1} = T_e \end{cases} \tag{5}$$

花费最小模型：

$$\min \quad \sum_{i=0}^{n} Cost_i$$
$$s.t \quad \begin{cases} e_i \in E_i \\ e_i \in (T_i, T_{i+1}) \qquad i = 0,1,2...n \\ T_{n+1} = T_e \end{cases} \tag{6}$$

中转次数最少模型：

$$\min \quad n$$
$$s.t \quad \begin{cases} e_i \in E_i \\ e_i \in (T_i, T_{i+1}) \qquad i = 0,1,2...n \\ T_{n+1} = T_e \end{cases} \tag{7}$$

实际上，以上模型的相关度很高，结果会比较相似，可以进行一定程度的统一。（详情请见4.2.2A）

3.2.2 问题二，地铁线路加入时的模型

从题目看来，地铁的线路仅有两条，而且分别是往返和环形线路，中间在第12和第18号站点相交，于是地铁的站点之间是相互可达的，由题意知，无论换乘与否，地铁的花费均是3元，这大大简化了模型和求解。

由地铁站点的两两连通性，可以认为如果需要乘坐地铁，仅仅需要乘坐一次；（5.1将给出说明）

这样，对于同样的问题，我们可以分两种情况，乘坐地铁和不乘坐地铁来求路径，在两组路径中取最优路径。

设带’的符号为乘坐地铁的情况下的站点或路径的符号，可以将模型清楚的表示出来。比如时间最优模型，可以调整为：

$$\min \quad \sum_{i=0}^{n} Time_i' + (n-1)\times\lambda_x$$
$$s.t \quad \begin{cases} e_i' \in E_i' \\ e_i' \in (T_i', T_{i+1}') \qquad i = 0,1,2...n' \\ T_{n+1}' = T_e' \end{cases} \tag{8}$$

这样我们的最优路径为上述路径与问题一中的最优路径中的最优。

依次类推可得包含地铁的线路的其他模型。

3.2.3 问题三，包含公交、地铁和步行的模型

1)简化的必要性和三种简化方案

步行时间表实际上给出了任意两个站点之间均有连接的线路，可以生成很多新的路径，大大增加了路径选择的灵活性，同时也给模型的求解增加了困难，因此必须给出相应的简化方案。我们设计了以下三种简化方案：

A.基于步行距离的线路剔除

考虑到很多距离远的的步行路线实际上没有意义，随着距离的增加，人们将逐渐对

步行丧失兴趣，即人对步行的不满度增加，将之量化并设为：

$$\lambda_f = f(Dis(T_x, T_y)) \tag{9}$$

f实际上是游客步行所承受的代价，是一个单调递增的函数，在现实中，很少游客可以接受需要步行较长的路线，于是一般f的增长速度较快，也就是说我们可以根据统计得出f, 对于f过大的距离，我们认为步行是不可行的。

距离的值与步行时间t,步行速度成正比，即：

$$Dis(T_x, T_y) = \bar{V} \times t \tag{10}$$

在的步行速度相差不大的情况下，可以认定t就是表征距离的量，故我们将$t>t_m$的线路从可行线路中剔出。

B.基于一个假设的剔除

将各种可能尽行分类，可以分为以下两种情况：

1．用步行来代替部分公交线路；

2．用步行连接原本没有连接的站点；

由实际情况，我们有以下假设：

步行速度比公交车速度低很多，即使公交车只走一站路，并且此路径的两端均为中转节点，仍然比步行要快；

由此假设，我们在考虑时间最优的模型时，将不考虑用步行代替公汽线路和地铁线路。

C．基于步行距离的归并

这用到了网络拓扑的办法。

将公交网络抽象成拓扑性质的网络图，路径搜索时不同属性的边之间在节点处的连通对应实际通行的车辆。换车要在公交站点处进行，如上分析，可以把适当距离内可能换车的多个站点抽象成一个节点。

不同交通线路的站点空间分布情况较为复杂，现以两条不同交通线路为例来说明。如下图所示，可以分为以下几种情况：(a)站点位置完全重合；(b)两站点不重合，但在同一道路上紧邻；(c)在两条道路交叉口的各自道路上紧邻；(d)在同一道路上较大距离相邻(近邻)；(e)在两条道路交叉口的各自道路上近邻；(f)在两条不相交的道路上近邻。图表 1 站点位置

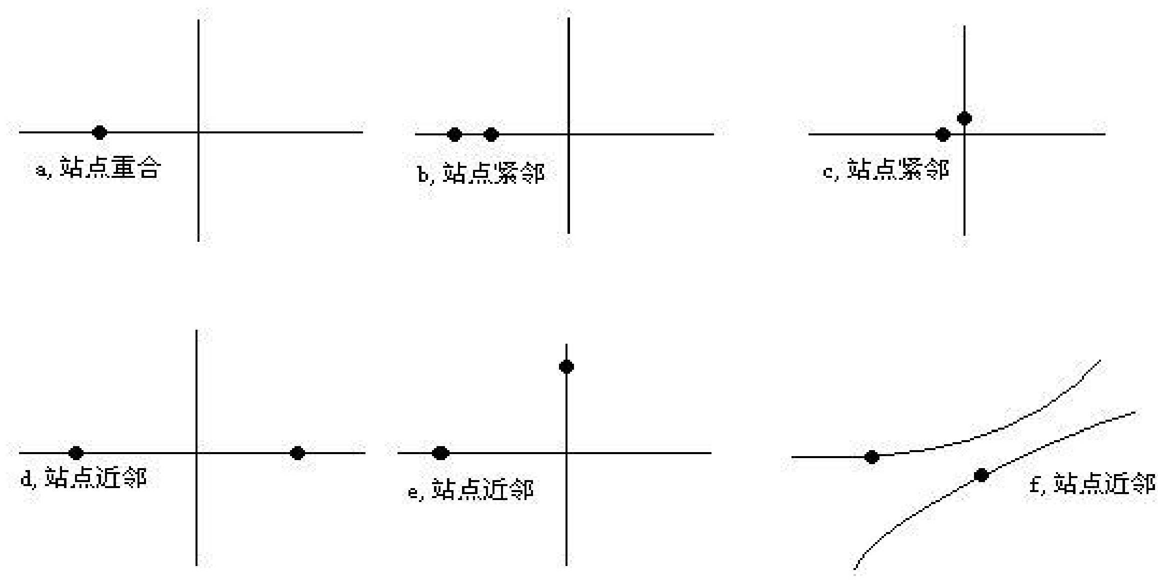


图 1 相邻的站点分类示意图

重合站点很容易处理，就是同一站点。紧邻站点和近邻站点的设置，目的是模拟人们在不同线路间的换车情况。把可以换车的不同线路的邻近站点抽象成同一节点，以利于路径寻优算法的实际计算。紧邻和近邻是两个半定量的距离概念，用以描述公交站点间空间位置上的距离关系。因此根据人们的行为习惯和平均公交路段长度，我们设置适当的缓冲区分析，不妨定义紧邻距离为：$d_i \le D_1$ ；近邻距离为：$D_1 \le d_i \le D_2$ 。是人为干预的经验值，其值和公交路段平均长度（耗时）呈线性相关。

紧邻站点和近邻站点的归并原则：（a）如果某一站点只有紧邻站点，那么该站点和它的紧邻站点归并。（b）如果某一站点没有紧邻站点，只有一个近邻站点，那么该站点和它的近邻站点归并。（c）如果有三个或更多个互为近邻的站点，这些站点可归并抽象为图上的一个节点。（d）某一站点 a 有两个近邻站点 b 和 c，而且 b 和 c 相互之间不

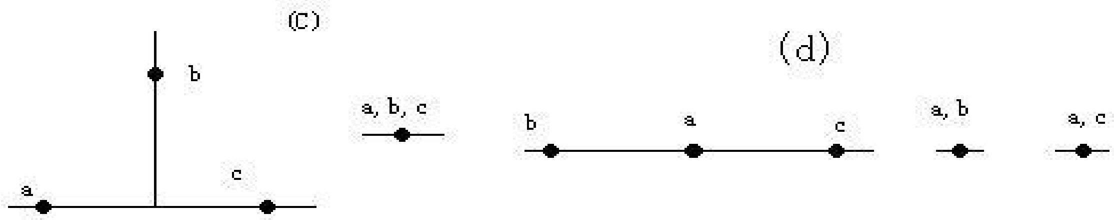


图 2 抽象原则一

是紧邻或近邻。我们采取的方法是把a同时归入b和c。（e）在归并站点时，同一节点不能包含大于近邻距离的两站点。（f）如图，当a、b、e、d四个站点两两相互近邻，但没有三个相互近邻时，可以把a、b归并c、d归并b、c归并，对应三个节点。其实这一原则是（d）和（f）的综合。

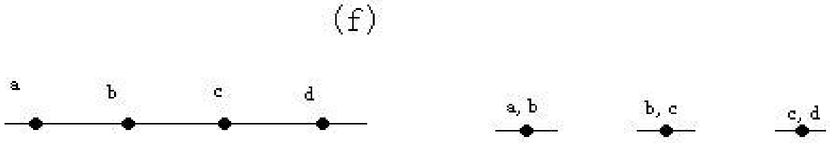


图 3 抽象原则二

一般的情况都可应用上述原则解决，在网络图节点的抽象过程中，可以由计算机模拟构造图。在数据收集和预处理时需进行人工干预，以取得较好的效果。[4]

我们的问题是已知两两站点间的步行时长且不考虑站点间除距离外的几何特性，所以我们只需定义步行时长不高于t_m的两个站点近（紧）邻。然后按以上原则将所有站点归并。这样，每个归并后的节点就会有更多的换乘选择，乘客就能用更少次的换乘到达终止站；而这也是一般乘客之所以选择步行的目的。

2)模型的建立

可以假设 T_i 被归入节点集 Tc_i'' 中。

经过如上的抽象节点的处理后，我们便可以再次进行已解决的公交线路中选路的过程。

优化方案的模型为：

不同要求下的分目标函数：

$$
\begin{aligned}
&\min \quad \sum_{i=0}^{n} \mathrm{Time}_i{}'' + \sum_{i=0}^{n} \tau i + (n-1)\times\lambda_x \\
&\min \quad \sum_{i=0}^{n} \mathrm{Cost}_i{}'' \\
&\min \quad m \\
&\text{s.t} \quad \begin{cases} e_i{}'' \in Ec_i{}'' \\ e_i{}'' \in (Tc_i{}'', Tc_{i+1}{}'') & i = 0,1,2\ldots n \\ T_e{}'' \in Tc''_{n+1} \\ \forall T_x \in Tc_i{}'', t(T_x, Tc_i{}'') < t_m \end{cases}
\end{aligned} \tag{11}
$$

其中带上标"的为对应模型一中的相应量， $t(T_x, T_v)$ 为步行经过对应两个节点的时间。

3) 一种评价的方法

可以简要的用步行时间 Time_1代替步行路程，或实际最短路径路程，用公交车的路径的时间Time_2来代替公交车或地铁线路的路程，这样，我们可以建立如下比值来评价公交线路的好坏。

$$\alpha = \frac{\mathrm{Time}_2}{\mathrm{Time}_1} \tag{12}$$

这样，α越大表示公交线路效率越低，往往代表着有不少类似往返的线路，于是，这种路经往往更值得步行优化。

4) 一个例子

下面给出一些完全基于假设步行时长的可行线路的特例以供参考：

以S3359→S1828为例，如果S3262与S3359间几何空间上的距离为徒步4分钟，我们就可以选择先徒步从S3359到S3262，然后搭乘L041路到达S1828。

同样的，也会有先搭乘公交到某一非终止点的站点，然后徒步走到目的站点的路线。如：S0008→S0073可选线路坐L200到S2534，从D15转地铁到D25后步行两站路到S0073。

中途换乘时换站一特例：S1557→S0481线路在S1157搭乘L084到其终点站S1919下车，徒步3站路走至S3186等L460路到S0481。

两端徒步特例：S0087→S3676选择徒步一站到S0630，乘L381到S0427，再徒步一站路到S3676。

如此可见，优化后的拓扑图又比以前有更好的性质。

4. 模型求解与结果

4.1 算法的要求

我们认定，算法应具备以下要求：

1.可靠性：给出任意两个站点，均要给出公交线路；

2.实用性：不必给出所有的路径，只要给出几条较好的路径即可；

3.实际性：不要给出换乘次数超过一定阈值的线路，即使这样的线路所用时间较少。因为在这种情况下，游客必须支付更多费用，并多次换乘；退一步讲，对于对时间要求较高的游客，估计会选择其他的交通方式，而不会走一条需要多次换乘的公交线路；
我们将从这3个方面来进行算法的评价；

4. 稳健性：即尽可能的给出乘坐不同交通线路的路径，这样某条线路堵车时，更可能找到备用路线。

4.2 模型一，仅考虑公交线路的模型

4.2.1 求解与结果

由于模型的搜索空间极大，如果简单用穷举的方法，是需要时间很长甚至无法实现的较多换乘次数的求解。我们的求解特色如下：

A．采用广度优先搜索

为了更快更好的求解模型一，我们采取广度搜索的原理，由起点得到e_0的集合，进而得到T_1 的集合,再得到e_1的集合，依次类推……

B．双向搜索

假设搜索到T_n的集合所用的时间为$O(m^n)$,可见我们不可能搜索过深；为了节省计算时间，采用双向搜索的算法，从起点和终点同时搜索，最终找到交集，这样我们所花费的时间 t满足:

$$t = 2 \times O(m^{n/2}) + t_s \tag{13}$$

t_s是判断从起点出发和到达终点的两个点集合的交集所用的时间；

由于t_s最多是两个集合均包含全部节点的时候对应的开销，大小为O(节点数^2)，当n较大时可以忽略，于是：

$$t = O(m^{n/2}) \tag{14}$$

也就是我们可以期望我们的算法比单向搜索节省相当多的时间；

但是，仍然不可能考虑太大的n,从实际出发，一个城市的公交线路应该满足对于绝大多数节点换乘2次即可抵达，结合4.1 中的实际性要求，乘公交需要转3次及其以上的路径，无论从最终结果还是游客感受都应该不会是最优的，这就意味着我们可以设定搜索总深度为2，即：

$n = 2$

C．动态规划与隐含枚举方法

为了进一步减少计算的复杂度，我们采取了动态规划和隐含枚举的方法：

所谓动态规划，是指整体最优由部分最优构成时采用每一部分求最优后再进行组合的方法；

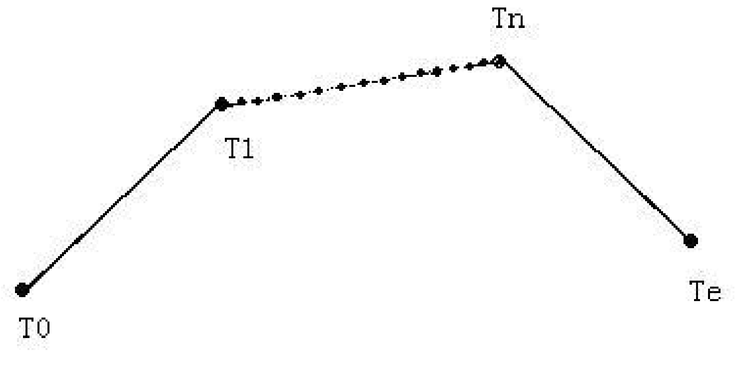


图 4 动态规划的理由

由上图可知，如果确定路径经过T1,T2,…Tn后，总体的最优（时间，花费）必然存在于个体最优中，假设每个个体的可行路径平均为m,则直接穷举的时间耗费：

$$t = O(m^n) \tag{15}$$

将每个路径军区最优后整合，时间耗费为：

$$t = n \times O(m) \tag{16}$$

所谓隐含枚举，就是指搜索最优路径时，如果多次搜到某一节点，我们将抛弃那些不符合目标条件的路径。

结合双向搜索，可以得到比较少的可行路径，最优解一定存在于这些可行路径的组合中。

为此我们可以用以下步骤进行求解：

第一步：数据读入与预处理

1） 读入数据并存入结构体line_station中，该结构体有520个元素，存储了txt 文件的所有信息，将char转化为int类型；

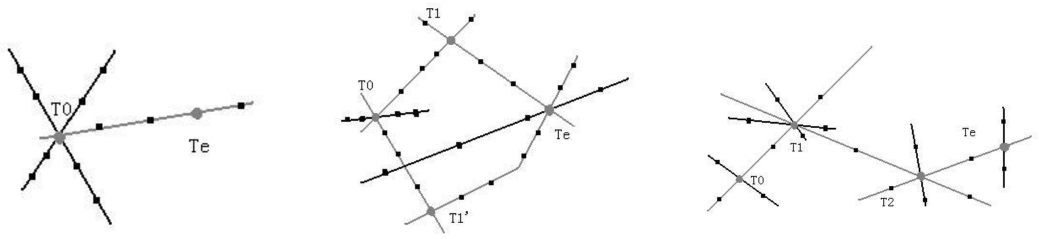
2） 将该结构体顺次扫描，可以得到各节点上的线路，存入另一个结构体station_line中。

3） 建立0-1矩阵，共有1040行，3959列，对于每一条线路，单行线反向生成返回路，上下行线分别记录于一行中，环路进行复制，对于3957个站点，如果线路i经过站点j，则矩阵的第i行j列为1，最后2列记录线路的票价和线路性质。

第二步：逐级寻解

根据第一步的结构体和矩阵可以很容易的建立两个函数，由Tx向下搜索后一节点Tx+1的函数f1(Tx)和由Ty向前搜索前一节点的函数f2(Ty),他们分别会生成两个节点集TRx和TBy，并保存最优路径;

根据输入的起止节点T_0,T_e，逐次比较生成路径；



(a) (b) (c)

图 5 搜索方法示意图

(灰色节点为当前路径选择的点，灰色线路为所经过的线路，类似黑色标示非路径)

如果$T_e \in TR_0$，那么两者之间不经过换乘就可以到达,如图2(a)；

如果$TR_0 \bigcap TB_e \neq \varnothing$，那么可以由$T_0$经过两点集的交集中的任意一个节点换乘到达$T_e$，经过遍历这些节点，可以给出一次换乘的最优解,如图2(b)。(其中$T_1{}'$为较差点，已经放弃)

如果$\exists T_1 \in TR_0, TR_1 \bigcap TB_e \neq \varnothing$，那么可以先由$T_0$经$T_1$再到交集中的任意节点换乘即可二次可达$T_e$，遍历每一个$T_1$下的交集，可以求出两次换乘的最优解,如图2(c)。

如果$\exists T_1 \in TR_0, \exists T_{n-1} \in TB_e, TR_0 \bigcap TB_e \neq \varnothing$，那么可以从$T_0$经$T_1$，再经交集中的任意节点，再经$T_{n-1}$可达$T_e$，类似遍历可得最优解,图示不再画出。

设定搜索节点的顺序为从距本站点最近的站点开始，可以期望首先得到较好的路径，结合动态规划和隐含枚举，减少搜索空间。

设定一定的最大搜索深度，比如2，可以把求解的时间控制在一定的范围内，但可能搜不到站点，如果出现这种情况，将有一个收尾算法采用非遍历算法求出以可行解。

第三步：中止准则（第二步没有成功时采用）

当不存在经历两个中转站点可达的路径时（这种情况非常少），我们将从远及近依次在T_1中选取一个节点作为新的起始站点，并记录更换首站信息以便最后恢复，重复第二步求经过新起始站点转二次车的路径，如果不成功，再次收尾，注意不能取曾经被取

为起始节点的站点为新的起始站点，就可以保证可以抵达任意节点。

最后，我们可以解得多组路线，从路线中找出满足三个子模型的最优的几条路径为：

表 1 模型一的多目标最优值列表

起 止 点	线路选择（箭头下方为经过的站点数）	花费/元	耗时/分钟	换乘次数
from:S3359 to: S1828	S3359 $\xrightarrow[31]{L436}$ S1784 $\xrightarrow[1]{L176}$ S1828	3	101	1
	S3359 $\xrightarrow[4]{L15}$ S1327 $\xrightarrow[13]{L201}$ S1790 $\xrightarrow[5]{L41}$ S1828	3	76	2
from:S1557 to: S0481	S1557 $\xrightarrow[12]{L084}$ S1919 $\xrightarrow[3]{L189}$ S3186 $\xrightarrow[17]{L460}$ S1481	3	106	2
	S1557 $\xrightarrow[12]{L363}$ S1919 $\xrightarrow[3]{L189}$ S3186 $\xrightarrow[17]{L460}$ S1481			
from:S0971 to: S0485	S0971 $\xrightarrow[20]{L013}$ S2184 $\xrightarrow[21]{L417}$ S0485	3	128	1
	S0971 $\xrightarrow[2]{L013}$ S1609 $\xrightarrow[19]{L140}$ S2654 $\xrightarrow[11]{L469}$ S0485	3	106	2
from:S0008 to: S0073	S0008 $\xrightarrow[14]{L463}$ S2083 $\xrightarrow[12]{L057}$ S0073	2	83	1
	S0008 $\xrightarrow[3]{L198}$ S3766 $\xrightarrow[13]{L296}$ S2184 $\xrightarrow[3]{L345}$ S0073	3	67	2
from:S0148 to: S0485	S0148 $\xrightarrow[14]{L308}$ S0036 $\xrightarrow[15]{L156}$ S2210 $\xrightarrow[3]{L471}$ S0485	3	106	2
	S0148 $\xrightarrow[14]{L308}$ S0036 $\xrightarrow[16]{L156}$ S3332 $\xrightarrow[2]{L471}$ S0485			
	S0148 $\xrightarrow[14]{L308}$ S0036 $\xrightarrow[17]{L156}$ S3351 $\xrightarrow[1]{L471}$ S0485			
from:S0087 to: S3676	S0087 $\xrightarrow[11]{L454}$ S3496 $\xrightarrow[9]{L209}$ S3676	2	65	1
	S0087 $\xrightarrow[1]{L21}$ S0088 $\xrightarrow[10]{L231}$ S0427 $\xrightarrow[1]{L097}$ S3676	3	46	2

（注：为了避免重复并使结果易于对比，我们把3个模型的求解结果列在一起，实际上以上各线路除第4，6组中转两次车的情况外，均已经达到费用最低；各组中的时间最少者，是在换乘2次及2次以内中的最小时间；各组中的最小换乘次数是最小换乘次数）

4.2.2 结果分析

A.三种模型的最优路径的重合问题

从上表中可以看出，很多组路径同时满足两个以上的最优，这到底是一种必然还是偶然？

我们的回答是，在搜索深度不大的情况下，三者的最优有较大的重合倾向。

换乘次数直接与费用相关，很显然每一次乘车都将付出相应的费用，换乘2次比换乘1次的期望费用一般不会降低，除非换乘一次的两段路径均为分段计价且道路比较长，换乘两次的道路均比较短或为单一票价。这种情况比较少。

换乘次数与运行时间存在一定的联系，但是这种联系较弱。

在换乘次数较少的情况，把换乘0，1，2……次的路径求其最短时间，可以得到最

优时间路径，同时必然有最少换乘，在相同换乘次数下，最短时间意味着最少路经节点，往往也是最优费用路径之一。

B．增加中转站点数目换取更少时间的原因及分析

由上表可以看出，经过较多两个中转节点可以一定程度上减少所用时间，这是因为在这种情况下，可能会少走一些额外的路线。

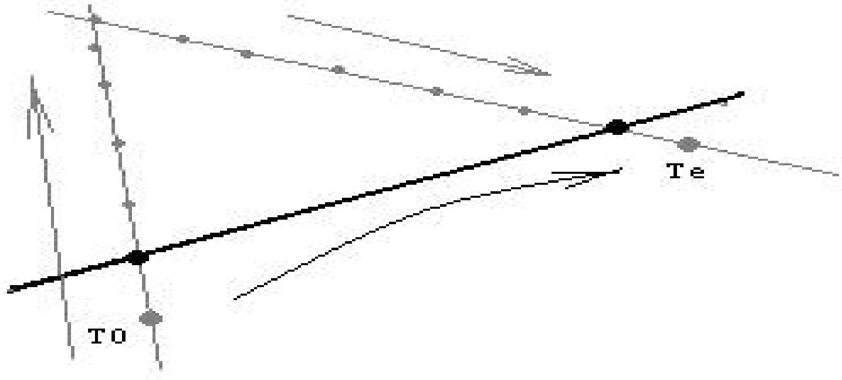


图 6 增加中转节点使总时间变短的典型情况

这样说明如果继续增加中转次数，可能会进一步减少所耗时间。但是，常识告诉我们，这样意义不大，因为随着中转次数的增多，转换站点浪费的时间增多，换乘带来的麻烦也随之增大。试想为了节省这为数不多的时间，选择换乘多次公交车，付出了很多的换乘花销，这往往是得不偿失的（可以考虑用打的来代替）。经过随机算法进行试验，我们发现两两之间存在中转次数不大于2的路径的概率大于95%，于是n=2 是可取的。

C．回顾4个要求并验证

1.可靠性，如果由起始站点可达的节点集 T_i 和可达终点的节点集 T_{n+1-j} 中站点总3957，则由抽屉原理，必有交集，可以肯定可以在i+j次中转后到达；当二者的站点数之和小于3957时，它们以一定的概率存在非空交集，上面已经提到，两两之间存在中转次数不大于2的路径的概率大于95%，于是我们可以肯定，经过并不多的试探，就可以得到多次换乘的路径；

2.实用性，我们在逐级求解的过程中，每次均剔除大量的不太好的路径，于是最终的可行路径空间不是很大，将其进一步精简，可以保证不至于给出太多的解；

3.实际性，广度搜索算法提供了换乘次数最少的路径，很好的满足了实际需要；

4.稳健性，我们尽量保留不同的线路，一条线路堵塞后，仍然很有希望从我们的线路中找出备用的路线；

总之，我们的模型较好的实现了以上的要求。

4.3 模型二，考虑地铁线路的模型

4.3.1 求解与结果

模型二加入了地铁，我们将路径分为两种情形:

情形一：经过地铁;

情形二：不经过地铁;

对于情形一，我们可以直接利用问题一的结果，从而简化模型二的工作量。

对于情形二，输入起点，终点后，由5.1的分析，可知最优的解只经过一次地铁，我们的实现过程如下：

第一步：通过对找到所有可以抵达地铁站点的公交站点；

第二步：利用模型一的求解中的算法，找到从起点到任意可乘地铁的站点的中转站尽可能少的线路，如果起点直接下地铁最好。

第三步：找到可达终点的以可乘地铁的站点的中转站尽可能少的线路，如果中点就是可乘地铁的线路最好。

第四步：进行枚举，得出最优路径。

我们的结果如下：

表 2 乘坐地铁的最优线路

起止点	线路选择（箭头下方为经过的站点数）	花费/元	耗时/分钟
from:S3359 to: S1828	$S3359\xrightarrow[5]{L015}S3068\rightarrow D8\xrightarrow[4]{T1}D12\xrightarrow[5]{T2}D38\rightarrow S3262\xrightarrow[10]{L041}S1828$	5	84.5
from:S1557 to: S0481	$S1557\xrightarrow[12]{L084/363}S1919\rightarrow D20\xrightarrow[2]{T1}D18\xrightarrow[9]{T2}D24\rightarrow S0537\xrightarrow[13]{L516}S0481$	5	119.5
from:S0971 to: S0485	$S0971\xrightarrow[6]{L094/119}S0567\rightarrow D1\xrightarrow[20]{T1}D21\rightarrow S0464/0466\xrightarrow[5]{L051/450/104/395/469}S0485$	5	96
from:S0008 to: S0073	$S0008\xrightarrow[6]{L200}S2534\rightarrow D15\xrightarrow[3]{T1}D12\xrightarrow[2]{T2}D25\rightarrow S0525\xrightarrow[2]{L103}S0073$	5	53.5
from:S0148 to: S0485	$S0148\xrightarrow[4]{L024}S1487\rightarrow D2\xrightarrow[19]{T1}D21\rightarrow S0464/S0466\xrightarrow[5]{L051/450/104/395/469}S0485$	5	87.5
from:S0087 to: S3676	$S0087\rightarrow D27\xrightarrow[8]{T2}D36\rightarrow S3676$	3	20

与纯公汽各项的最优线路相比：

S3359→S1828 坐地铁会浪费 8.5 分钟，最少花费多 2 元钱和 2 次换乘；

S1557→S0481 坐地铁不但要浪费 13.5 分钟，最少花费多 2 元钱和 1 次换乘；

S0971→S0485 坐地铁节省 10 分钟，最少花费多 2 元钱和 1 次换乘；

S0008→S0073 坐地铁节省 13.5 分钟，最少花费多 3 元钱和 2 次换乘；

S0148→S0485 坐地铁节省 18.5 分钟，最少花费 2 元钱；

S0087→S3676 坐地铁节省 26 分钟，最少花费多 1 元钱但可少换乘一次。

4.3.2 结果分析

由以上分析，我们可以看出，对于大多数路径，，走地铁比不走地铁线路要节省时间，因为地铁速度快，但是也有例外。比如

$S3359\rightarrow S1828$ 线路，终点站距离地铁出口较远，而且地铁中间要换乘。

$S1557\rightarrow S0481$ 线路，起点终点距离出站口远，乘地铁中途仍要换乘。

最好的线路 $S0087\rightarrow S3676$ 是很好的，经过地铁可以直达。

一般而言，起点终点距离地铁较远的线路，可能经过地铁并不划算。

5. 针对问题的两点讨论

5.1 不会两次乘坐地铁的理由

地铁系统的示意图如下：

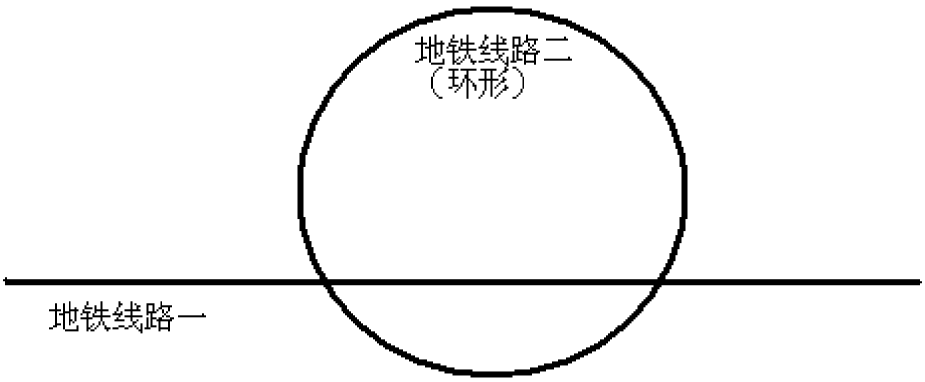


图 7 地铁线路示意图

其中交点为 12、18 号站点。

如果两次乘坐地铁，那么我们实际上只是相当于从地铁的一个站点 A 到另外的一个站点 B。显然不可能节省费用和转乘次数；由于实际的地铁线路二是双环，顺时针和逆时针均有，于是，在地铁较之公交车快的情况下，无论 A,B 均处于线路一还是均处于线路二，都不应该中间转公共汽车，而增加的转车的花销有 14-4=10 分钟，是地铁行驶 4 站的时间，在地铁具有较好的拓扑结构时，是不可能比一直坐地铁节省时间的。

综上所述，我们只需考虑乘坐一次地铁的情形。

5.2 我们对查询系统的改进意见

我们认为，成功的公交查询模型应该有以下特点：

考虑多种公交工具：对照模型二，模型一的现实意义稍小，对于游客而言，一般情况下，将地铁一起考虑提供的方案更好，这就启发我们，如果可以考虑更多的交通方式（如的士，船只甚至缆车），应该可以得到更为优秀的方案，不过这样使问题的规模成倍增大。

存储备查：查询系统应该将尽可能多组起始站点的最优路径存储起来，这样可以避免很多计算，提高反应速度，如果有困难，可以对于一些客流量较多的站点存储其最优路径信息。

动态分析：可以即时的对整个城市的公交状况进行采样和分析，使游客尽量通过避免拥挤的路径，同时也使城市的交通流量趋于平均，缓解交通压力。

智能搜索：在做这个程序时，尽管采用了多种方法优化算法，但是仍然存在大量的无效搜索，可以引入智能搜索系统，将待选线路进行初步筛选，（比如反向行驶的线路基本不应该考虑），可望大大减少搜索空间，将搜索向深度发展从而更快更好的给出建议路径。

6. 模型的评价与推广

本模型采用了多种方法有效的减少计算量，使计算的时间减少到可以接受的程度，大部分情况在数秒钟内可以得出结果。

但是模型仍然可以进一步优化，由于时间仓促，水平有限，我们没能将这些想法完

全实现, 现将想法整理如下：

鉴于搜索路径的数目很大，在没有实际的站点地理坐标图中，可以将节点进行聚类分析，由实际公交车道路的布局，可以知道相距较近的节点有较多的共同道路，相距较远的节点又较少的共同道路，于是我们可以将节点聚类，对于不在一类的节点，如果不能直达，我们应该优先搜索可以到达终点所在聚类的那些路径，这样实际上是更可能向着目标站点的方向前进。

可以综合考虑 3 种模型，将目标函数进行极差标准化，并且可以在实际中通过对乘客的调查一组比较合适的λ_i，进而可以得到一个加权的目标：

$$\min \quad \lambda_1 \times \frac{TotalTime}{TotalTime_{max} - TotalTime_{min}} + \lambda_2 \times \frac{TotalCost}{TotalCost_{max} - TotalCost_{min}} + \lambda_3 \times \frac{n}{n_{max} - n_{min}} \tag{16}$$

（其中$\sum_{i=1}^{3} \lambda_i = 1$）

在具体的查询系统中可以按照用户的要求给出3种模型的最优值和综合模型的最优值，从而使用户得到需要的最佳路线。

最后一个模型参照实际公交线站点空间关系分析，依据已知信息将原本很复杂的公交线路拓扑为新的节点更少，路线选择更灵活的交通网路图，优化使得原本会花很多时间且会产生较差结果的算法可以在新的拓扑图中产生更好的效果。这种基于实际情况合理优化对象已是同一方法能达更好效果的思想也是解决实际问题中一种非常重要的思想。

我们讨论问题一直没有牵扯到站点间的空间几何特性，作为一个十分现实的模型，公司也应该调查取样将站点的空间几何特性的信息录入其数据库，并在此基础上，如上面的站点抽象的讨论那样将北京所有公交站点拓扑结构化。如此，这个模型就更加具有实际意义了。

参考文献

[1] 谭浩强，C++程序设计，北京：清华大学出版社，2004

[2] 钱颂迪，甘应爱，田丰等，运筹学，北京：清华大学出版社，1990

[3] 殷剑宏，吴开亚，图论及其算法，合肥：中国科学技术出版社，2003

[4] 陆忠，钱翔东，张登荣.基于最短路径查询的城市公交网络拓扑建模研究.遥感信息，2002，(1):11-14

[5] 李季涛，杨俊锋. 基于GIS的公交网络模型及其在公交线路查询中的应用.大连铁道学院学报，2004，25(2):30-33